Сравнительная оценка вариантов реализации модели обработки данных ASP в корпоративных веб-приложениях

Comparative evaluation of options for implementing the ASP data processing model in corporate web applications

Саенко / Saenko I.
Игорь Борисович
(ibsaen@mail.ru)
доктор технических наук, профессор.
ФГБУН Санкт-Петербургский институт информатики и автоматизации Российской академии наук,
ведущий научный сотрудник.
г. Санкт-Петербург

Ясинский / Yasinsky S.
Сергей Александрович
(yasinsky777@mail.ru)
доктор технических наук, доцент.
ЗАО «Институт телекоммуникаций»,
ведущий специалист.
г. Санкт-Петербург

Шаповалов / Shapovalov D.
Дмитрий Сергеевич
(dimah1996@mail.ru)
ФГКВОУ ВО «Военная академия связи им. Маршала Советского Союза С. М. Буденного» МО РФ, курсант.
г. Санкт-Петербург

Ключевые слова: информационная система – information system; модель обработки данных – data processing model; веб-сервер – web server; веб-технологии – web technologies; активные серверные страницы – active server pages.

Рассматривается модель обработки данных на основе технологии «активных серверных страниц», используемая для разработки веб-приложений в корпоративных информационных системах. Приводится описание возможных вариантов реализации этой модели. Обсуждаются результаты сравнительной оценки вариантов реализации модели обработки данных ASP в корпоративных веб-приложениях.

The data processing model based on the technology of "active server pages" used for the development of web applications in corporate information systems is considered. The description of possible options for implementing this model is given. The results of the comparative evaluation of options for implementing the ASP data processing model in corporate web applications are discussed.

Введение

В настоящее время в корпоративных информационных системах (КИС) достаточно острым является вопрос сбора, обработки, хранения и выдачи разнородной информации, что объясняется резким увеличением размеров поступающих в базы данных (БД) информационных потоков и необходимостью обеспечения высокого качества пользовательских информационных услуг. Дальнейшее развитие КИС связано с ростом многозадачности процессов управления, увеличением количества хостов и уровней распределенности обработки информации, а также лавинообразным ростом объемов хранимой в БД информации, приводящим к необходимости решения проблемы Больших Данных [1, 2]. Одним из направлений преодоления этих вызовов является активное использование при разработке КИС современных технологий разработки веб-приложений [3].

К средствам разработки корпоративных веб-приложений в настоящее время предъявляются различные требования, к которым относятся, в частности, требования поддержки полного функционала общеязыковой среды исполнения, полноты инструментария, быстрого понимания основ разработки программистом и высокой скорости исполнения проекта [4]. Одними из наиболее распространенных средств разработки веб-приложений, в полной мере отвечающими этим требованиям, являются средства реализации технологии «активных серверных страниц» (англ. Active Server Pages – ASP). ASP-технология реализует возможность создания веб-страниц, выступающих в качестве исполнителей сценариев и, тем самым, обеспечивает независимость от типа клиентского браузера [5–7]. Считается, что ASP-технология не является защищенной, однако она может использоваться для оценки рисков безопасности [8], а при ее реализации не в среде Windows, а например в AstraLinux [9] этот ее недостаток можно не учитывать. Все это делает ASP-технологию доста-

2-2019.indd 72 14.06.2019 16:59:51

точно привлекательной [10]. В то же время, возможны различные варианты реализации ASP-технологии, и перед разработчиками веб-приложений зачастую стоит вопрос выбора наиболее приемлемого из них. Поэтому целью настоящей работы является разработка модели обработки данных в КИС на основе ASP-технологии, определение возможных вариантов ее реализации и сравнительная оценка этих вариантов.

Модель обработки данных ASP в КИС

Общая последовательность функционирования «активных серверных страниц» в КИС выглядит следующим образом (рис. 1).

Клиентом является конечный пользователь, работающий с сервером КИС со своего рабочего места (РМ). В терминологии рассматриваемой модели он является должностным лицом (ДЛ). Клиент с удаленного РМ ДЛ посылает запрос на ASP-страницу, хранящуюся на сервере КИС. Сервер принимает запрос и приступает к его обработке.

Обработка запроса начинается с определения запрашиваемого файла с расширением *.asp, после чего сервер работает с содержимым этого файла через обработчик. Обработчик ASP интерпретирует и выполняет вставки ASP-кода, содержащего обращения к источникам данных. Источником данных в рассматриваемом на рис. 1 случае являются элементы SqlDataSourse, в которых прописываются соответствующие строки подключения к БД, элементы структур запросов для выбора (SELECT), вставки (INSERT), обновления (UPDATE) и удаления (DELETE) данных. Примерами таких вставок являются %Response.Write% (внесение изменений в набор данных, возвращаемых клиенту) и %Data()% (определение даты).

Таким образом, на сторону РМ ДЛ отправляется гипертекстовая HTML-страница без ASP-кода. Исходя из этого, можно сделать вывод, что копирование серверных сценариев клиентом невозможно, так как он получает лишь результаты их выполнения [7].

Варианты реализации модели обработки данных ASP

Рассмотренная выше модель обработки данных ASP имеет четыре возможных варианта своей реализации, представленные в таблице 1. Эти варианты определяются режимами использования двух возможных способов контроля данных: валидации, осуществляющей контроль вводимых данных, и специально созданного класса, предназначенного для замены элемента управления (контрола).

В варианте «ВК» используется валидация и специальный класс. В варианте «НН» не используется ни валидация, ни специальный класс. В вариантах «НК» и «ВН», соответственно, либо не используется валидация, но используется, либо валидация используется, а специальный класс – нет.

Валидация, являющаяся первой составляющей модели обработки данных ASP, играет важную роль при оптимизации работы многих сложных проектов. Однако в некоторых случаях рациональнее применять и другие варианты обеспечения эффективной работы приложения. В частности, в БД КИС для контроля ввода данных зачастую используются выпадающие списки, в которых клиент может выбирать символы по умолчанию для их ввода в БД. После попадания таких символов в БД возможно возникновение ошибки из-за несоответствия установленного разработчиком типа данных символьному типу. В итоге клиенту предо-

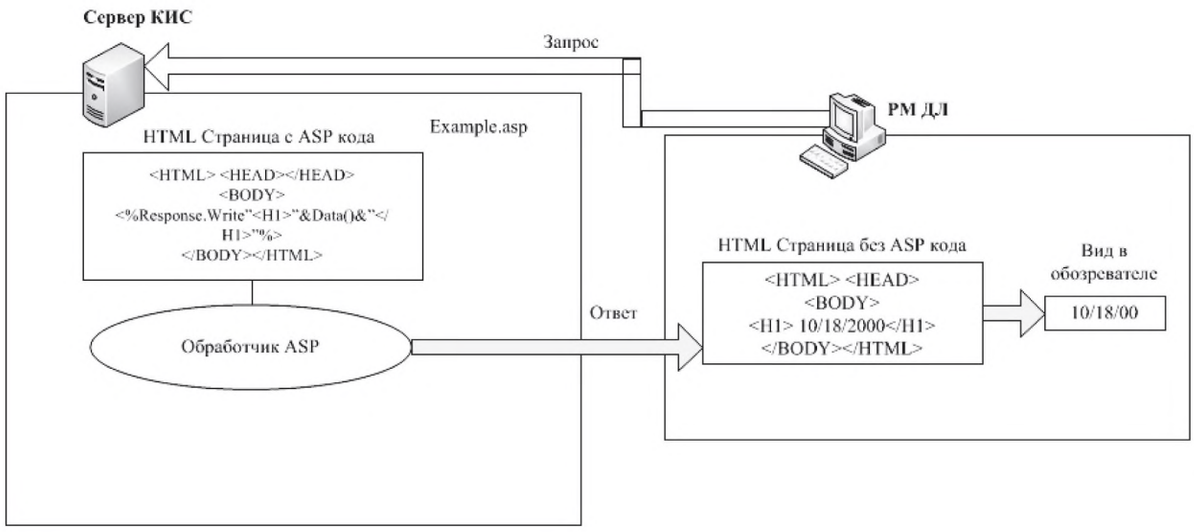

Рис. 1. Модель обработки данных ASP в КИС

2-2019.indd 73 14.06.2019 16:50:54

Таблица 1

Варианты реализации модели обработки данных ASP

Режим использования валидации	Режим использования специального класса	
	Используется	Не используется
Используется	ВК	НК
Не используется	ВН	НН

ставляется некорректное отображение текущих и ранее введенных данных. В некоторых случаях, при наличии дополнительных исключений со стороны SQL-запросов, возможна безвозвратная потеря информации. Данная проблема решается установкой флага запрета на отображение некорректного символа. Благодаря этому выбор некорректного элемента INSERT-запроса становится невозможным.

Так как контролы содержат скрытые поля ViewState для поддержки собственного состояния при учете изменений на стороне клиента, то циркулирующий между запросами трафик в случае использования этих элементов управления будет иметь повышенный объем. Исходя из этого, возникает идея отображения заголовка веб-страницы без использования меток либо каких-то дополнительных контролов (не берется за основу даже Literal, который не имеет HTML-аналога), снижая тем самым загруженность сети. Для реализации данной идеи создается следующий специальный класс Head [11]:

```
public class Head
{public string Text { get; set;}
public Head (string text)
{this.Text = text;}}
```

В папке программы App_Data размещается текстовый файл TextFile.txt. Это делается с целью, чтобы исключить доступ к файлам со стороны клиента. Подобным образом осуществляется построение крупных проектов, исключая возможность копирования серверных сценариев. При этом опытные разработчики также используют уникальные идентификаторы для каждого файла в определенной кодировке.

Содержимое этого файла имеет следующий вид:

```
<div>
<p><%REG%></p>
</div>
```

Код для преобразования виртуального адреса файла в физический добавляется в папку Page_Load, после чего выполняется следующая замена:

```
string filename = Server.MapPath(@"App_Data\TextFile.txt");
output = File.ReadAllText(filename);
Head h = new Head("Регистрация");
output = output.Replace("<%REG%>", h.Text);
```

Заголовок в разметке выводится при помощи кодовой вставки <%=output%>.

Экспериментальная оценка вариантов реализации модели

Экспериментальная оценка предложенных вариантов реализации модели ASP проводилась в среде веб-приложения, разработанного для информационной системы учета результатов научной работы, функционирующей в Военной академии связи (http://vas.mil.ru/Nauka). Трафик, который необходимо фиксировать в этой системе, имеет разнородных характер за счет наличия большого количества различных органов управления (ОУ). Кроме того, возникает необходимость перекрестного учета информации при взаимодействии различных ОУ и ДЛ друг с другом. Таким образом, эта система в полной мере относится к классу распределенных систем обработки информации и представляет собой достаточно характерный пример современной и развитой КИС.

Стартовое меню разработанного приложения представлено на рис. 2. В нем имеется три закладки: «Регистрация», «Добавить работу» и «Назначить работу». Реализация обработки данных при использовании этих закладок осуществлялась в соответствии с возможными вариантами модели ASP.

Оценка эффективности работы веб-приложения, реализующего модель ASP, производилась на электронной вычислительной машине следующей конфигурации: Intel(R) Pentium(R) CPU B960 @ 2.20 GHz 2.20 GHz, ОЗУ 4 Гб, ОС Windows Server 2012 R2 Standard, x64, Visual Studio 2010 версии 10.0.30319.1 RTMRel, Microsoft.NET Framework версии 4.5.52282 RTMRel 2010. Для отображения результатов оценки эффективности работы веб-приложения использовалась программа «Монитор ресурсов» версии 6.3.9600. Интервал измерений в каждом эксперименте составлял 60 секунд.

В таблице 2 представлены результаты оценки вариантов реализации модели ASP по объему загружаемых наборов данных.

Из анализа этих результатов можно сделать вывод, что применение рассмотренных выше вариантов реализации модели ASP существенно отражается на загрузке процессов локального веб-сервера и среды конфигурирования. При этом прослеживается высокая динамика изменения объема следующих видов ресурсов: зарезервированного операционной системой под процесс, используемого под процесс, используе-

2-2019.indd 74 14.06.2019 16:50:54

Рис. 2. Стартовое меню веб-приложения

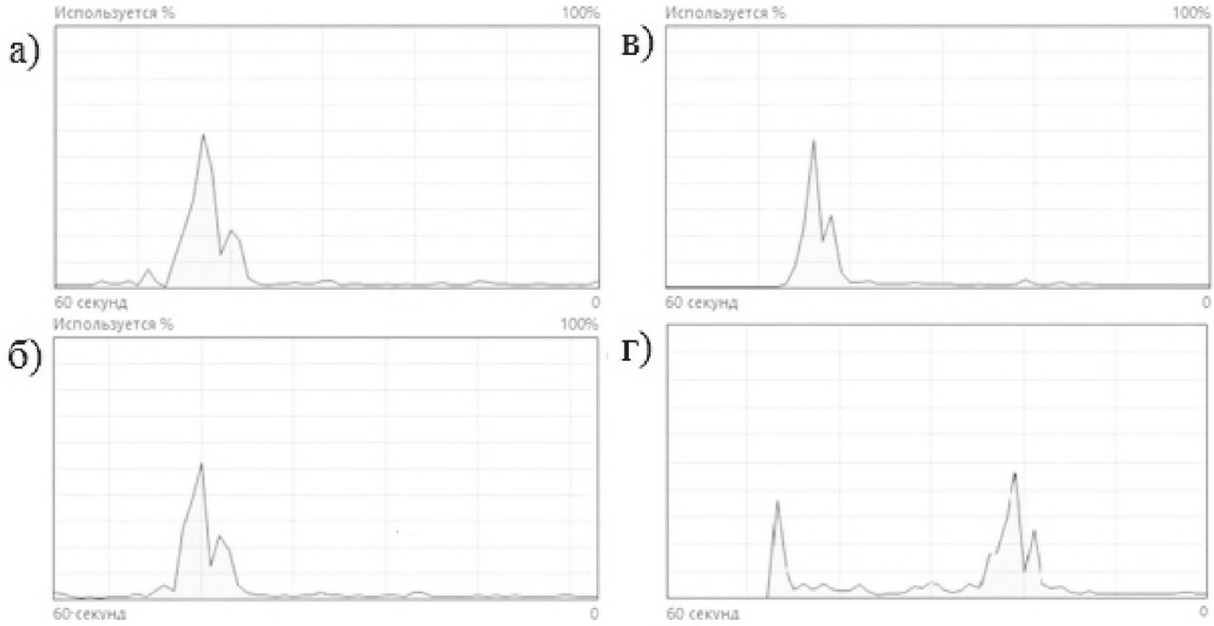

Рис. 3. Графики загрузки ЦП: а) вариант «ВК», б) вариант «НН», в) вариант «НК», г) вариант «ВН»

2-2019.indd 75 14.06.2019 16:50:55

Таблица 2

Оценка объемов загружаемых наборов данных

Процесс	Вариант	Загружаемый набор данных, Кб			
		Выделение	Рабочий	Общий	Частный
sqlservr.exe	ВК	71524	62196	22528	39668
postgres.exe		3608	17844	15708	2136
WebDev.WebServer40.exe		38692	52856	34924	17932
sqlservr.exe	НН	71528	62040	22521	39519
postgres.exe		3540	17788	15708	2080
WebDev.WebServer40.exe		38816	53188	35124	18064
sqlservr.exe	НК	71524	62192	22528	39664
postgres.exe		3608	17844	15708	2136
WebDev.WebServer40.exe		38676	52928	34980	17948
sqlservr.exe	ВН	71528	62048	22524	39524
postgres.exe		3540	17788	15708	2080
WebDev.WebServer40.exe		38976	53352	35160	18192

мого (неиспользуемого) совместно с другими процессами. На рисунке 3 представлены графики загрузки центрального процессора (ЦП) при различных способах реализации модели, снятые программой «Монитор ресурсов».

Анализ приведенных выше графических зависимостей показал, что применение валидации уместно при наличии небольшого приложения (варианты «ВК» и «ВН»). Что касается больших приложений, то необходимо внимательно изучать структуру проекта, чтобы найти эффективные способы их замены, где это возможно. Использование варианта с применением классов в обход стандартных элементов управления (варианты «ВК» и «НК»), приводит к появлению скачков в загрузке ЦП, без этих классов (остальные варианты) – к концентрированной загрузке. Следовательно, выбор наиболее предпочтительного варианта реализации модели обработки данных ASP зависит от многих факторов, в том числе – от конкретной задачи, решаемой в КИС. С точки зрения безопасности наиболее интересным представляется вариант «ВК», так как для него интервал загрузки системы небольшой. В этом случае злоумышленник будет не в состоянии за столь малый промежуток времени внести вредоносный трафик. Варианты «ВК» и «НН» имеют значительные отличия по всем параметрам каждого из рассматриваемых ключевых процессов. Вариант «НН» рассматривается только для сравнения, так как в реальных проектах построения КИС он не применяется.

Следует также заметить, что на фоне продолжающегося развития компьютерных технологий необходимы соответствующие улучшения не только со стороны программного обеспечения, но и со стороны дорогостоящего компьютерного оборудования. Проведенные эксперименты показывают возможность и необходимость изменения параметров новых или уже функционирующих КИС. Так, для анализа процесса, отвечающего за исполняемый модуль веб-сервера WebDev.WebServer40.exe, необходимо рассматривать возможность реализации вариантов «ВН» и «НК». При применении варианта «НК» этот процесс, в отличие от «ВН», имеет меньшую ресурсоемкость, что позволяет внедрять его аналоги в системы с малой производительностью, снижая, тем самым, затраты вычислительных ресурсов.

Заключение

Проведенный сравнительный анализ возможных вариантов практической реализации модели обработки данных в БД КИС на основе ASP-технологии показал, что в зависимости от тех или иных условий необходимо использовать определенный способ реализации этой модели. Правильно выбранный вариант реализации модели ASP будет напрямую влиять на работу веб-сервера КИС, среду конфигурирования, управления, доступа и разработки его компонентов, сочетающую в себе элементы Analysis Manager, Enterprise Manager и Query Analysis. Кроме того, с учетом связи данных процессов со службами (Reporting Services, Integration Services и другими), большую значимость приобретает приоритет анализа и учета предложенных вариантов реализации модели ASP на всех этапах разработки корпоративных веб-приложений.

Дальнейшие направления исследований связываются с интеграцией модели ASP в модель веб-сервера, а также с анализом и разработкой наиболее эффективных подходов к реализации этой модели на базе Unix-подобных операционных систем путем использования коллекции библиотек.

Работа выполнена при поддержке РФФИ (проект 18-07-01369) и бюджетной темы № АААА-А16-116033110102-5.

2-2019.indd 76 14.06.2019 16:59:55

Литература

1. Gärtner B. Issues with Big Data / B. Gärtner, M.R.W. Hiebl // The Routledge Companion to Accounting Information Systems. Martin Quinn and Erik Strauß (Eds.). – Chapter 13. – 2018. – P. 1–16.

2. Котенко, И. В. Архитектура системы параллельной обработки больших данных для мониторинга безопасности сетей Интернета вещей / И.В. Котенко, И.Б. Саенко, А.Г. Кушнеревич // Труды СПИИРАН. – 2018. – № 4(59). – С. 5–30.

3. Котенко, И. В. Перспективные системы хранения данных для мониторинга и управления безопасностью информации / И.В. Котенко, И.Б. Саенко, О.В. Полубелова // Труды СПИИРАН. – 2013. – № 2 (25). – С. 113–134.

4. Botwe D.A. A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software / D.A. Botwe, J.G. Davis // International Journal of Computer Science and Mobile Computing. – 2015. – Vol. 4, Issue 2. – P. 154–165.

5. Беллиньясо М. Разработка веб-приложений в среде ASP.NET 2.0: задача – проект – решение / М. Беллиньясо. – М.: «Диалектика», 2007.

6. Мак-Дональд М., Шпушта М. Microsoft ASP.NET 2.0 с примерами на C# 2005 для профессионалов. – М.: «Вильямс», 2007.

7. Экспозито Д. Знакомство с технологией Microsoft ASP.NET 2.0 AJAX. – СПб: Питер, 2007. – 321 с.

8. Evaluation of information security risks using hybrid assessment model / B.K. Alese [et al.] // Proceedings of the 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014). – 2014.

9. Безопасность операционной системы специального назначения Astra Linux Special Edition / В.П. Буренин [и др.] – М.: Горячая линия – Телеком, 2016. – 312 с.

10. Осипов, Н. А. Разработка приложений ASP.NET с применением Entity Framework / Н.А. Осипов. – СПб: Университет ИТМО, 2016. – 80 с.

11. Кровчик, Э. Сетевое программирование для профессионалов / Э. Кровчик, В. Кумар, Н. Лапари. – М.: «Лори», 2007. – 417 с.

2-2019.indd – 77 14.06.2019 – 16:50:56